

Proyecto: **Tienda de Magia**

Lenguaje: **Python**

Framework: **Django**

Editor: **VS Code**

1. Procedimiento para crear carpeta del Proyecto:

Crear una carpeta llamada **UIII\_TiendadeMagia\_0533**

2. Procedimiento para abrir VS Code sobre la carpeta:

Abrir VS Code, seleccionar la opción "Abrir carpeta" y elegir **UIII\_TiendadeMagia\_0533**.

3. Procedimiento para **abrir terminal en VS Code**:

**Presionar las teclas Ctrl + ñ o ir a Ver → Terminal.**

4. Procedimiento para crear carpeta de entorno virtual **".venv"** desde la terminal de VS Code:

```
python -m venv .venv
```

5. **Procedimiento para activar el entorno virtual:**

```
.venv\Scripts\activate
```

6. Procedimiento para **activar intérprete de Python**:

En VS Code, presionar Ctrl + Shift + P, escribir Python: Select Interpreter, y elegir el intérprete que diga **.venv**.

7. Procedimiento para instalar **Django**:

```
pip install django
```

8. Procedimiento para crear proyecto **backend\_TiendadeMagia** sin duplicar carpeta:

```
django-admin startproject backend_TiendadeMagia .
```

9. Procedimiento para ejecutar servidor en el **puerto 8007**:

```
python manage.py runserver 8007
```

10. Procedimiento para copiar y pegar el link en el navegador:

Copiar el enlace que aparece en la terminal (<http://127.0.0.1:8007/>) y pegarlo en el navegador.

11. Procedimiento para crear aplicación **app\_TiendadeMagia**:

```
python manage.py startapp app_TiendadeMagia
```

12. Aquí el modelo models.py:

```
from django.db import models
```

```
# =====
```

```
# MODELO: PRODUCTOS
```

```
# =====
```

```
class Producto(models.Model):
```

```
    nombre = models.CharField(max_length=100)
```

```
    descripcion = models.TextField(blank=True, null=True)
```

```
    categoria = models.CharField(max_length=50)
```

```
    precio = models.DecimalField(max_digits=10, decimal_places=2)
```

```
    proveedor = models.CharField(max_length=100)
```

```
    fecha_ingreso = models.DateField(auto_now_add=True)
```

```
    stock = models.IntegerField()
```

```
    def __str__(self):
```

```
        return self.nombre
```

```
# =====
```

```
# MODELO: ORDEN DE VENTA
```

```
# =====
```

```
class OrdenDeVenta(models.Model):
```

```
    cliente = models.CharField(max_length=100)
```

```
    fecha_orden = models.DateTimeField(auto_now_add=True)
```

```
    direccion_envio = models.CharField(max_length=200)
```

```
    metodo_pago = models.CharField(max_length=50)
```

```
    total = models.DecimalField(max_digits=10, decimal_places=2)
```

```
    estado = models.CharField(max_length=50)
```

```
    comentarios = models.TextField(blank=True, null=True)
```

```
# Relación muchos a muchos con Productos
```

```
productos = models.ManyToManyField(Producto, related_name="ordenes")
```

```
    def __str__(self):
```

```
        return f"Orden #{self.id} - {self.cliente}"
```

```
# =====
# MODELO: DETALLE ORDEN
# =====
class DetalleOrden(models.Model):
    orden = models.ForeignKey(OrdenDeVenta, on_delete=models.CASCADE,
related_name="detalles")
    producto = models.ForeignKey(Producto, on_delete=models.CASCADE,
related_name="detalles")
    cantidad = models.IntegerField()
    precio_unitario = models.DecimalField(max_digits=10, decimal_places=2)
    subtotal = models.DecimalField(max_digits=10, decimal_places=2)
    descuento = models.DecimalField(max_digits=10, decimal_places=2, default=0)
    observaciones = models.CharField(max_length=200, blank=True, null=True)

    def __str__(self):
        return f"{self.producto.nombre} x {self.cantidad}"
```

12.5 Procedimiento para realizar las migraciones (makemigrations y migrate):

```
python manage.py makemigrations
python manage.py migrate
```

13. Primero trabajamos con el MODELO: PRODUCTO

14. En views.py de **app\_TiendadeMagia** crear las funciones: inicio\_tienda, agregar\_producto, actualizar\_producto, realizar\_actualizacion\_producto, borrar\_producto.

15. Crear la carpeta “**templates**” dentro de “**app\_TiendadeMagia**”.

16. En la carpeta templates crear los archivos html (**base.html, header.html, navbar.html, footer.html, inicio.html**).

17. En base.html **agregar Bootstrap** para CSS y JS.

18. En navbar.html incluir las opciones:

“Sistema de Administración Tienda de Magia”, “Inicio”, “Productos”, “Órdenes”, “Detalles de Órdenes”.

19. En **footer.html** incluir derechos de autor, fecha del sistema y “Creado por Omar Carrasco, Cbtis 128”.

20. En inicio.html colocar información del sistema y una imagen de magia.

21. Crear subcarpeta “**producto**” dentro de **app\_TiendadeMagia/templates**.

22. Crear los archivos html (agregar\_producto.html, ver\_productos.html, actualizar\_producto.html, borrar\_producto.html).

23. No utilizar [forms.py](#).

24. Crear archivo [urls.py](#) en **app\_TiendadeMagia**.

25. Agregar **app\_TiendadeMagia** en settings.py de **backend\_TiendadeMagia**.

26. Configurar [urls.py](#) de **backend\_TiendadeMagia** enlazar con **app\_TiendadeMagia**.

**27. Registrar modelos en [admin.py](#) y volver a realizar las migraciones.**

```
from django.contrib import admin
from .models import Producto, OrdenVenta, DetalleOrden
admin.site.register(Producto)
admin.site.register(OrdenVenta)
admin.site.register(DetalleOrden)
```

28. Por lo pronto solo trabajar con "Producto".

29. Utilizar colores suaves, atractivos y modernos.

30. No validar entrada de datos.

31. **Crear estructura completa** de carpetas y archivos.

32. Proyecto totalmente funcional.

33. Finalmente ejecutar el servidor en el **puerto 8007**:

```
python manage.py runserver 8007
```